Appl. No. 09/992,642

R E M A R K S

Claims 3, 18, 26 and 27 have been amended. Support for the amendments to claims 3 and 26 is found at least in originally filed claim 9. Support for the amendments to claims 18 and 27 is found at least in the summary. No new matter has been added.

Claims 1, 2 and 9 have been cancelled. Claims 4-8, 10-17 and 19-25 stand as originally filed.

Claims 1-27 were considered in the Office Action. Claims 1, 3, 5-9, 11, 18, 26 and 27 stand rejected under 35 U.S.C. 102(e) as being anticipated by Buzbee, U.S. Patent 6,219,832. Claims 2, 4 and 20 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Buzbee in view of Porter, U.S. Patent 6,357,040. Claims 10 and 12-17 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Buzbee in view of Blaise et al., U.S. Patent 6,505,344 (hereinafter referred to as Blaise). Claim 19 stands rejected under 35 U.S.C. 103(a) as being unpatentable over Buzbee in view of Bugnion, U.S. Patent 6,704,925. Claims 21-25 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Buzbee in view of Ng, U.S. Patent 6,704,314.

The Declaration is objected to as missing a statement acknowledging the duty to "disclose to the Office all information known to the person to be material to patentability as defined in §1.56". The originally filed Declaration does contain this statement. A copy of the originally filed Declaration is submitted herewith with this statement underlined for convenience.

Applicants believe that the currently pending claims are not anticipated by or obvious over the cited references for at least the reasons set forth below and respectfully request reconsideration.

The Invention of Claim 3

The cited references do not disclose or suggest:

"A method of enabling dynamic optimization of a computer program, comprising:

generating annotation information about said computer program, **said annotation information being derived from information held by a compiler about references to individual memory locations**; and

storing said annotation information with said computer program, said annotation information enabling a dynamic optimizer to optimize said computer program during execution."

(Claim 3, as amended, emphasis added)

The above highlighted features are not anticipated or suggested by the cited references and would not have been obvious to a person with ordinary skill in the art having the cited references. Buzbee's annotations describe the **number of formal parameters that are declared for a procedure**. (Buzbee, col. 4, lines 45-47) Applicants respectfully note that Buzbee's annotations are not directed at entry points, but at the number of formal parameters declared for a procedure or entry point. Even if Buzbee's annotations were directed at entry points, this would not be information about **references** to individual memory locations. Buzbee's number of formal parameters declared for a procedure does not suggest or anticipate the claimed annotation information that is derived from information about references to individual memory locations.

Applicants therefore believe that claim 3 is allowable over the cited references and respectfully request

reconsideration.

Dependent claims 4-7 depend upon independent claim 3 which is allowable over the cited art as discussed above. These dependent claims are likewise in condition for allowance at least because they depend on an allowable independent claim. However, dependent claims 4-7 are independently allowable at least in that they recite particular features which, when combined with the elements of the independent claim, are also not disclosed or suggested in the cited references.

Claim 8 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

> "The method of claim 3, wherein said generating annotation information comprises generating annotation information **derived from runtime architecture and software conventions.**"
> (Claim 8, emphasis added)

Again, Buzbee's annotations are directed at the number of formal parameters declared for a function. The number of formal parameters declared is independent of runtime architecture.

Claim 10 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

> "The method of claim 3, wherein said generating annotation information comprises generating **annotation information identifying a unique stack pointer register to be used by said computer program.**"
> (Claim 10, emphasis added)

Blaise has been cited as disclosing "inline stack locations as an annotation." (Office Action mailed December 22, 2004, paragraph 8) Applicants respectfully disagree with this characterization, and further believe that even if this characterization were true, it would not disclose or suggest the limitations of claim 10. Blaise generates "inline stacks" based on the actual parameters and return values of call sites, forming a "list of methods that must be inlined at particular call sites in order for a given allocation instruction to allocate objects on the stack, rather than on the heap." (Blaise, col. 9, lines 12-20) This does not disclose or suggest "identifying a unique stack pointer register".

Blaise is also not directed at dynamic optimization or at annotations stored with an executable program, but at the building of a live call graph and escape analysis during code generation. (Blaise, col. 7, lines 56-67) Applicants respectfully disagree that there is any motivation to combine Blaise and Buzbee, or that even if combined, the result would read on Applicants' claim 10.

Claim 11 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

> "The method of claim 3, wherein said generating annotation information comprises generating **annotation information comprising a list of non-ambiguous memory locations.**"
> (Claim 11, emphasis added)

Although Buzbee's annotations are directed to the number of formal parameters declared for a procedure, they do not comprise a "list of non-ambiguous memory locations". The

number of formal parameters is not a list of memory locations. Furthermore, Buzbee does not disclose the format of references to memory locations and therefore does not disclose that the references are in the form of non-ambiguous memory locations. The term "non-ambiguous memory location" is defined in Applicants' specification at page 13, last paragraph, as follows: "[E]ach such memory reference instruction individually must reference the exact same memory location each time it is executed." Buzbee does not disclose or suggest that references to memory locations in annotations are non-ambiguous. (As indicated above, Buzbee does not even include references to memory locations in annotations.)

Claim 12 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

> "The method of claim 11, **wherein said annotation information enables said dynamic optimizer to obtain canonical names for said non-ambiguous memory locations.**" (Claim 12, emphasis added)

The references (including Buzbee and Blaise) do not discuss the format of references to memory locations, and therefore do not disclose or suggest enabling a dynamic optimizer to obtain canonical names for non-ambiguous memory locations. Buzbee is directed to the number of formal parameters declared for a procedure, Blaise is directed to parameters and return values. Neither discusses the format of references to memory locations.

Claim 13 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

"The method of claim 11, **wherein said non-ambiguous memory locations comprise stack frame locations.**"
(Claim 13, emphasis added)

Again, neither Buzbee nor Blaise disclose or suggest annotations directed to references to non-ambiguous memory locations, nor non-ambiguous stack frame locations.

Claim 14 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

"The method of claim 3, wherein said generating annotation information comprises generating **annotation information comprising a mapping of memory references to all non-ambiguous locations which are referenced.**"
(Claim 14, emphasis added)

Neither Buzbee nor Blaise disclose or suggest a mapping of memory references to all non-ambiguous locations which are referenced.

Claim 15 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

"The method of claim 3, wherein said generating annotation information comprises generating **annotation information comprising a list of canonical names of stack frame locations that are promotable.**"
(Claim 15, emphasis added)

Neither Buzbee nor Blaise disclose or suggest a list of canonical names of stack frame locations that are promotable.

Claim 16 is believed allowable as depending from an

allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

"The method of claim 3, wherein said generating annotation information comprises generating **annotation information comprising a guarantee that no stack frame location is live beyond the scope of the stack frame.**" (Claim 16, emphasis added)

Neither Buzbee nor Blaise disclose or suggest a guarantee that no stack frame location is live beyond the scope of the stack frame.

Claim 17 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

"The method of claim 3, wherein said generating annotation information comprises generating **annotation information comprising a format and a location of stack unwinding information.**" (Claim 17, emphasis added)

Neither Buzbee nor Blaise disclose or suggest annotation information comprising a format and a location of stack unwinding information.

The Invention of Claim 18

The cited references do not disclose or suggest:

"A method of dynamically optimizing a computer program, comprising:

reading annotation information **derived from runtime architecture and software conventions used to compile**

said computer program, said annotation information being stored with said computer program; and

dynamically optimizing said computer program based on said annotation information while said computer program is being executed."

(Claim 18, as amended, emphasis added)

As discussed above, the cited references do not disclose or suggest annotation information derived from runtime architecture and software conventions. For example, Buzbee is directed to the number of formal parameters declared for a procedure, Blaise is directed to parameters and return values. Neither of these is dependent upon runtime architecture or software conventions.

Dependent claim 19 depends upon independent claim 18 which is allowable over the cited art as discussed above. This dependent claim is likewise in condition for allowance at least because it depends on an allowable independent claim. However, dependent claim 19 is independently allowable at least in that it recites particular features which, when combined with the elements of the independent claim, are also not disclosed or suggested in the cited references.

Claim 20 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

"The method of claim 18, wherein said **dynamically optimizing said computer program comprises replacing subroutine calls in said computer program with inline program code.**"

(Claim 20, emphasis added)

Although Buzbee does disclose dynamic optimization based

on annotations, and Porter does disclose inlining frequently accessed functions, neither discloses the generation of annotations enabling the inlining of functions by dynamic optimization at runtime. Buzbee's annotations do not provide the information needed to dynamically inline subroutines. Porter discloses optimization while building software, and so has no need to obtain information from an external source. The combination of Buzbee and Porter would therefore not be capable of dynamically replacing subroutine calls with inline program code based on the disclosed contents of the annotations.

Claim 21 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

> "The method of claim 18, wherein said dynamically optimizing said computer program comprises **removing redundant callee-save register restores.**"
> (Claim 21, emphasis added)

Although Ng does address removing redundancy using an optimizing compiler, Ng specifies two types of redundancies: fully and partially redundant expressions which have been either fully or partially evaluated previously in paths leading to the expression. (Ng, col. 1, lines 40-47) Optimizing out redundant expressions that have been previously evaluated does not disclose or suggest the removal of redundant callee-save register restores as in Applicants' claim 21. Applicants respectfully disagree that a general statement that redundant expressions may be optimized away discloses or suggests the specific limitation of claim 21.

Claim 22 is believed allowable as depending from an allowable base claim and is further believed allowable in that

the cited references do not disclose or suggest:

> "The method of claim 18, wherein said dynamically optimizing said computer program comprises **propagating constant arguments within said computer program.**" (Claim 22, emphasis added)

Applicants repeat the arguments for allowability set forth above with respect to claim 21, but specifically directed to the method set forth in claim 22. None of the cited references disclose or suggest the propagation of constant arguments while dynamically optimizing a computer program.

Claim 23 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

> "The method of claim 18, wherein said dynamically optimizing said computer program comprises **promoting local data from a stack frame location to a register.**" (Claim 23, emphasis added)

Applicants repeat the arguments for allowability set forth above with respect to claim 21, but specifically directed to the method set forth in claim 23. None of the cited references disclose or suggest promoting local data from a stack frame location to a register while dynamically optimizing a computer program.

Claim 24 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

> "The method of claim 18, wherein said dynamically

optimizing said computer program comprises **removing redundant callee register saves.**"
(Claim 24, emphasis added)

Applicants repeat the arguments for allowability set forth above with respect to claim 21, but specifically directed to the method set forth in claim 24. None of the cited references disclose or suggest removing redundant callee register saves while dynamically optimizing a computer program.

Claim 25 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

"The method of claim 18, wherein said dynamically optimizing said computer program comprises **removing stack frame allocation.**"
(Claim 25, emphasis added)

Applicants repeat the arguments for allowability set forth above with respect to claim 21, but specifically directed to the method set forth in claim 25. None of the cited references disclose or suggest removing stack frame allocation while dynamically optimizing a computer program.

The Invention of Claim 26

The cited references do not disclose or suggest:

"Apparatus for enabling dynamic optimization of a computer program, the apparatus comprising:

one or more computer readable storage media; and

computer executable instructions stored in the one or more computer readable storage media, the computer

executable instructions comprising:

instructions for generating annotation information about said computer program, wherein said annotation information enables a dynamic optimizer to optimize said computer program during execution, **said annotation information being derived from information held by a compiler about references to individual memory locations**; and

instructions for storing said annotation information with said computer program."

(Claim 26, as amended, emphasis added)

Applicants repeat the arguments for allowability set forth above with respect to claim 1, but specifically directed to the apparatus set forth in claim 26. None of the references disclose or suggest annotation information being derived from information held by a compiler about references to individual memory locations.

The Invention of Claim 27

The cited references do not disclose or suggest:

"Apparatus for dynamically optimizing a computer program, the apparatus comprising:

one or more computer readable storage media; and

computer executable instructions stored in the one or more computer readable storage media, the computer executable instructions comprising:

instructions for reading **annotation information derived from runtime architecture and software conventions used to compile said computer program**, said annotation information being stored with said computer program; and

instructions for dynamically optimizing said computer program based on said annotation information while said computer program is being executed."

(Claim 27, as amended, emphasis added)

Applicants repeat the arguments for allowability set forth above with respect to claim 8, but specifically directed to the apparatus set forth in claim 27. None of the references disclose or suggest annotation information derived from runtime architecture and software conventions.

In view of the above, all of the claims are believed to be in condition for allowance, and the Applicants respectfully request that a timely Notice of Allowance be issued.

Dated: 3/1/05

Respectfully submitted,
KLAAS, LAW, O'MEARA & MALKIN, P.C.

By: _____

Guy R. Clinger, Esq.
Registration No. 42,422
1999 Broadway, Suite 2225
Denver, CO 80202
(303) 298-9888
Fax: (303) 297-2266